**培训要求**

**1）掌握JSON及其应用**

**2）了解jQuery的背景和特点**

**3）理解js对象和jQuery对象的区别**

**4）掌握jQuery九类选择器及应用（上）**

**声明：今天服务端我们使用Struts2技术**

**一）什么是JSON**

（1）JSON（Java Script Object Notation（记号,标记））是一种轻量级的数据交换语言，

以文本字符串为基础，且易于让人阅读

注意：XML就是一个重量级的数据交换语言

（2）JSON采用完全独立于任何程序语言的文本格式，使JSON成为理想的数据交换语言

**二）JSON的作用**

（1）简化创建自定义对象的方式

注意：JSON就是用JS语法来书写，所以必须放在<script>标签中

在用JS语法书写JSON时，最外面不要用""双引号

使用json 可以简化创建

练习 一

|  |
| --- |
| <!-- js方式定义Person对象  <script type="text/javascript">  function Person(id,name,sal){  this.id = id;  this.name = name;  this.sal = sal;  }  var p = new Person(1,"波波",7000);  document.write("编号:" + p.id + "<br/>");  document.write("姓名:" + p.name + "<br/>");  document.write("薪水:" + p.sal + "<br/>");  </script>  -->  <hr/>  <!-- json方式定义Person对象 -->  <script type=*"text/javascript"*>  //采用js语言来书写  **var** p = {id:1,name:"包包",sal:8000};    //属性可以''或""符号  //字符串必加''或""符号，其它类型不用加符号  document.write("编号:" + p.id + "<br/>");  document.write("姓名:" + p.name + "<br/>");  document.write("薪水:" + p.sal + "<br/>");  </script> |

练习 二 一个 { } 代表一个对象

|  |
| --- |
| <script type=*"text/javascript"*>  **var** ps = [  {  id:1,  name:'哈哈'  },  {  id:2,  name:'呵呵'  },  {  id:3,  name:'嘻嘻'  }  ];  </script>    <script type=*"text/javascript"*>  document.write("共有" + ps.length + "个学生<br/>");  **for**(**var** i=0;i<ps.length;i++){  document.write("编号：" + ps[i].id + "<br/>");  document.write("姓名：" + ps[i].name + "<br/>");  }  </script> |

练习 三 属性的 [ ] 为数组型 数组内可以有: “ ”字符串 、 { } 对象 ….

|  |
| --- |
| var p = {'city':['北京','上海','广州','深圳']};  for(var i=0;i<p.city.length;i++){  document.write(p.city[i]+"<br/>");  } |

（2）在AJAX中，作为数据载体之一

注意：JS可以直接解析JSON格式的文本，前提是：该JSON必须采用JS格式书写的才行，如果该JSON是采用Java格式写的，必须使用eval()函数转换后，方可被JS解析，该eval("")函数接收一个字符串格式的内容。

（3）省份-城市-区域三级联动【Struts2 + JSON版】

切记：将来JSON是不能完完全全替代XML的，只能在定义对象和数据交换语言方面替代

JSP

|  |
| --- |
| <head>  <title>省份-城市-区域三级联动【Struts2 + JSON版】</title>  <script type=*"text/javascript"* src=*"js/ajax.js"*></script>  </head>  <body>    <select id=*"provinceID"* style="width:*111px*">  <option>选择省份</option>  <option>湖北</option>  <option>湖南</option>  <option>广东</option>  </select>    <select id=*"cityID"* style="width:*111px*">  <option>选择城市</option>  </select>    <select id=*"areaID"* style="width:*111px*">  <option>选择区域</option>  </select>    <!-- 省份->城市 -->  <script type=*"text/javascript"*>  //定位下拉框  document.getElementById("provinceID").onchange = **function**(){  //清空城市下拉框  **var** cityElement = document.getElementById("cityID");  cityElement.options.length = 1;  //清空区域下拉框  **var** areaElement = document.getElementById("areaID");  areaElement.options.length = 1;    **var** province = this[this.selectedIndex].innerHTML;  **if**("选择省份" != province){  //-1- 调用上面的创建对象  **var** ajax = createAJAX();  //-2- 准备发送请求  **var** method = "POST";// 有参数传入 且有汉字 findCityByProvinceRequest 是xml中的 name  **var** url = "${pageContext.request.contextPath}/findCityByProvinceRequest?time="+**new** Date().getTime();  ajax.open(method,url);  //-3-  ajax.setRequestHeader("content-type","application/x-www-form-urlencoded")  //-4- bean 是Action 中的名字 ，province 是 Bean类中的属性  **var** content = "bean.province=" + province;  ajax.send(content);    //-------------------------------------------等待    //NO5  ajax.onreadystatechange = **function**(){  **if**(ajax.readyState == 4){  **if**(ajax.status == 200){  //-6- 接收 java格式的json  **var** jsonJAVA = ajax.responseText;    //jsonJAVA 是 java格式的json文本 不能别直接执行  //解决方案 ： 将java格式的json文本 转成js格式的 json文本  // 用js 提供的一个函数可以搞定 固定格式： jsonJava两边加上括号  **var** jsonJS = eval("("+jsonJAVA+")");    **var** array = jsonJS.cityList;  **var** size = array.length;  **for**(**var** i=0;i<size;i++){  **var** city = array[i];  **var** option = document.createElement("option");  option.innerHTML = city;  cityElement.appendChild(option);  }  }  }  }  }  }  </script>  <!-- 城市->区域 -->  <script type=*"text/javascript"*>  document.getElementById("cityID").onchange = **function**(){  **var** areaElement = document.getElementById("areaID");  areaElement.options.length = 1;  **var** city = this[this.selectedIndex].innerHTML;  **if**("选择城市" != city){  //NO1)  **var** ajax = createAJAX();  //NO2)  **var** method = "POST";  **var** url = "${pageContext.request.contextPath}/findAreaByCityRequest?time="+**new** Date().getTime();  ajax.open(method,url);  //NO3)  ajax.setRequestHeader("content-type","application/x-www-form-urlencoded")  //NO4)  **var** content = "bean.city=" + city;  ajax.send(content);    //------------------------------------------等待    //NO5)  ajax.onreadystatechange = **function**(){    **if**(ajax.readyState == 4){  **if**(ajax.status == 200){  //NO6)  **var** jsonJAVA = ajax.responseText;  **var** jsonJS = eval("("+jsonJAVA+")");  **var** array = jsonJS.areaList;  **var** size = array.length;  **for**(**var** i=0;i<size;i++){  **var** area = array[i];  **var** option = document.createElement("option");  option.innerHTML = area;  areaElement.appendChild(option);  }  }  }  }    }  }  </script>  </body> |

action:

|  |
| --- |
| Bean类：  province 、 city 、 有get()、set()、 Bean(){} 方法  省份-城市-区域三级联动【Struts2 + JSON版】  **public** **class** ProvinceCityAreaAction **extends** ActionSupport{  **private** Bean bean; //新建一个Bean类 统一封装对象  **public** Bean getBean() {  **return** bean;  }  **public** **void** setBean(Bean bean) {  **this**.bean = bean;  }    //根据省份获取城市  **public** String findCityByProvince() **throws** Exception {  cityList = **new** ArrayList<String>();  **if**("湖北".equals(bean.getProvince())){  cityList.add("武汉");  cityList.add("赤壁");  }**else** **if**("湖南".equals(bean.getProvince())){  cityList.add("郴州");  cityList.add("张家界");  cityList.add("浏阳");  }**else** **if**("广东".equals(bean.getProvince())){  cityList.add("阳江");  cityList.add("清远");  cityList.add("佛山");  cityList.add("湛江");  }  //让struts2框架帮你将List/Set/Map<String>转成JSON文本  **return** ***SUCCESS***;  }  // 根据城市获取区域  **public** String findAreaByCity() **throws** Exception {  areaList = **new** ArrayList<String>();  **if**("阳江".equals(bean.getCity())){  areaList.add("AA");  areaList.add("BB");  }**else** **if**("清远".equals(bean.getCity())){  areaList.add("CC");  areaList.add("DD");;  }**else** **if**("佛山".equals(bean.getCity())){  areaList.add("EE");  areaList.add("FF");  }**else** **if**("湛江".equals(bean.getCity())){  areaList.add("GG");  areaList.add("HH");  }  **return** ***SUCCESS***;  }  //配置完xml后 做成实例变量  **private** List<String> areaList;//区域的集合  **private** List<String> cityList;//城市的集合  //json 会扫描是否有get方法 有会把 获取到的值 以json文本返回  **public** List<String> getCityList() {  **return** cityList;  }  **public** List<String> getAreaList() {  **return** areaList;  }  } |

struts.xml

|  |
| --- |
| <struts><!--  1.添加struts2-json-plugin 架包  2.多继承 : json-default 本身继承了struts-default  3.加上result 一定要有type="json" 不能转发重定向  -->  <package name=*"myPackage"* extends=*"json-default"* namespace=*"/"*>    <!-- 全局结果类型 -->  <global-results>  <result name=*"success"* type=*"json"*/>  </global-results>    <!-- 省份->城市 -->  <action  name=*"findCityByProvinceRequest"*  class=*"cn.itcast.javaee.js.provincecityarea.ProvinceCityAreaAction"*  method=*"findCityByProvince"*>  </action>  <!-- 城市->区域 -->  <action  name=*"findAreaByCityRequest"*  class=*"cn.itcast.javaee.js.provincecityarea.ProvinceCityAreaAction"*  method=*"findAreaByCity"*>  </action>  </package>  </struts> |

导入：

|  |
| --- |
| struts2-json-plugin-2.3.1.1.jar |

**三）使用第三方工具，将JavaBean对象/List或Set或Map对象转成JSON**

准备导入第三方jar包：

》commons-beanutils-1.7.0.jar

》commons-collections-3.1.jar

》commons-lang-2.5.jar

》commons-logging-1.1.1.jar

》ezmorph-1.0.3.jar

》json-lib-2.1-jdk15.jar

（1）JavaBean----->JSON

》JSONArray jsonArray = JSONArray.fromObject(city);

》String jsonJAVA = jsonArray.toString();

（2）List<JavaBean>----->JSON

》JSONArray jsonArray = JSONArray.fromObject(cityList);

》String jsonJAVA = jsonArray.toString();

（3）List<String>----->JSON

》JSONArray jsonArray = JSONArray.fromObject(stringList);

》String jsonJAVA = jsonArray.toString();

（4）Set<JavaBean>----->JSON

》JSONArray jsonArray = JSONArray.fromObject(citySet);

》String jsonJAVA = jsonArray.toString();

（5）Map<String,Object>----->JSON

》JSONArray jsonArray = JSONArray.fromObject(map);

》String jsonJAVA = jsonArray.toString();

最后一个例子切记，将来jQuery-EasyUI-DataGrid组件时我们还要用到

将来，在企业中，就算脱离struts2的环境，也能用第三方工具，将Java类型转成JSON文本

|  |
| --- |
| **public** **class** TestBean2Json {  **private** **static** **void** javabean2json() {  City city = **new** City(1,"广州");  JSONArray jSONArray = JSONArray.*fromObject*(city);  String jsonJAVA = jSONArray.toString();  System.***out***.println(jsonJAVA);  //[{"id":1,"name":"广州"}]  }  **private** **static** **void** list2json() {  List<City> cityList = **new** ArrayList<City>();  cityList.add(**new** City(1,"广州"));  cityList.add(**new** City(2,"珠海"));  JSONArray jSONArray = JSONArray.*fromObject*(cityList);  String jsonJAVA = jSONArray.toString();  System.***out***.println(jsonJAVA);  //[{"id":1,"name":"广州"},{"id":2,"name":"珠海"}]  }  **private** **static** **void** set2json() {  Set<City> citySet = **new** LinkedHashSet<City>();  citySet.add(**new** City(1,"广州"));  citySet.add(**new** City(2,"珠海"));  JSONArray jSONArray = JSONArray.*fromObject*(citySet);  String jsonJAVA = jSONArray.toString();  System.***out***.println(jsonJAVA);  //[{"id":1,"name":"广州"},{"id":2,"name":"珠海"}]  }  **private** **static** **void** javabeanlist2json() {  List<City> cityList = **new** ArrayList<City>();  cityList.add(**new** City(1,"中山"));  cityList.add(**new** City(2,"佛山"));  Province province = **new** Province(1,"广东",cityList);    JSONArray jSONArray = JSONArray.*fromObject*(province);  String jsonJAVA = jSONArray.toString();  System.***out***.println(jsonJAVA);  /\*  [  {  "id":1,  "name":"广东"  "cityList":[{"id":1,"name":"中山"},{"id":2,"name":"佛山"}],  }  ]  \*/  }  **private** **static** **void** map2json() {    List<City> cityList = **new** ArrayList<City>();  cityList.add(**new** City(1,"中山"));  cityList.add(**new** City(2,"佛山"));    Map<String,Object> map = **new** LinkedHashMap<String,Object>();  map.put("total",cityList.size());//表示集合的长度  map.put("rows",cityList);//rows表示集合    JSONArray jSONArray = JSONArray.*fromObject*(map);  String jsonJAVA = jSONArray.toString();  System.***out***.println(jsonJAVA);  //[{"total":2,"rows":[{"id":1,"name":"中山"},{"id":2,"name":"佛山"}]}]    jsonJAVA = jsonJAVA.substring(1,jsonJAVA.length()-1); //去括号  System.***out***.println(jsonJAVA);  //{"total":2,"rows":[{"id":1,"name":"中山"},{"id":2,"name":"佛山"}]}  } |

**四）总结JSON的特点**

（1）在客户端（特指PC浏览器），直接使用JavaScript语言解析JSON，无需第三方jar包

（2）本质上，就是一个文本，只是该文本有特定的书写格式

（3）可以使用第三方工具，将JavaBean对象或者List/Set/Map<JavaBean>对象转成JSON

（4）优点：JSON与XML很相似，但是它更加轻巧，服务器只需发送一个html普通字符串，不用发送复杂的xml格式文档了

（5）缺点：语法过于严谨，初学者可能觉得代码不易读，写错一点都不行

（6）JSON本质上，就是用JS语法写的特殊文本记号，用JS可以直接解析

**五）模拟jQuery库，体验使用第三方实用库的特点【图片隐藏与显示】**

学会：JS的封装思想

学会：创建自定义对象和优化方法

学会：引用第三方实用的库，该库中预定义大量实用的对象和函数

学会：查阅第三方实用的库提供的API手册

|  |
| --- |
| //定义一个Photo函数，看作是一个类  function Photo(){  //属性  var imgElement = document.images[0];  //方法  this.show = function(){  imgElement.style.visibility = "visible";  }  this.hide = function(){  imgElement.style.visibility = "hidden";  }  }  //定义一个$()函数，用来定位标签  function $(str){  //如果str变量是字符串类型  if( typeof(str) == "string" ){  //获取str变量中的第一个字符  var init = str.substring(0,1);  //如果第一个字符是#的话  if("#" == init){  //获取str变量中除第一个字符外的其它字符  var other = str.substring(1,str.length);  //通过ID定位节点  var element = document.getElementById(other);  //如果找到了节点  if(element != null){  //返回  return element;  }else{  //返回  return null;  }  }else{  //继续判断  }  }else{  alert("参数必须为string类型");  }  }  <script type="text/javascript">  //创建一个Photo对象  var p = new Photo();  //定位隐藏按钮，同时添加单击事件  $("#hide").onclick = function(){  //调用Photo对象的方法  p.hide();  }  //定位显示按钮，同时添加单击事件  $("#show").onclick = function(){  //调用Photo对象的方法  p.show();  }  </script> |

**六）jQuery是什么**

John Resig在2006年1月发布的一款跨主流浏览器的JavaScript库，简化JavaScript对HTML操作

参见<<什么是jQuery.JPG>>

**七）为什么要使用jQuery**

（1）写少代码,做多事情【write less do more】

（2）免费，开源且轻量级的js库，容量很小

注意：项目中，提倡引用min版的js库

（3）兼容市面上主流浏览器，例如 IE，Firefox，Chrome

注意：jQuery不是将所有JS全部封装，只是有选择的封装

（4）能够处理HTML/JSP/XML、CSS、DOM、事件、实现动画效果，也能提供异步AJAX功能

（5）文档手册很全，很详细

（6）成熟的插件可供选择

（7）提倡对主要的html标签提供一个id属性，但不是必须的

（8）出错后，有一定的提示信息

（9）不用再在html里面通过<script>标签插入一大堆js来调用命令了

**八）jQuery开发步骤**

（1）引用第三方js库文件，<script type="text/javascript" src="js/jquery-1.8.2.js"></script>

（2）查阅并使用api手册，$("#divID").html()/val()/text()/css("color","red")/....

|  |
| --- |
| //var divElement = document.getElementById("divID");  var $div = $("#divID");  //var html = divElement.innerHTML;  var html = $div.html();  alert(html); |

**九）js对象和jQuery对象相互转换**

（1）什么是js对象及代码规则

就是使用js-API，即Node接口中的API或是传统JS语法定义的对象，叫做js对象

js代码规则----divElement

var divElement = document.getElementById("divID");

var nameArray = new Array(3);

（2）什么是jQuery对象及代码规则

就是使用jQuery-API，返回的对象就叫做jQuery对象

jQuery代码规则----$div

var $div = $("#divID")

声明：上述代码规则，只是老师个人规则，不代表所有企业都这样做

（3）js对象转成jQuery对象【重点】

语法：$(js对象)---->jQuery对象

例如：$(divElement)---->$div

例如：$(this)---->$this

注意：jQuery对象将js对象做了封装，js对象二边无引号

|  |
| --- |
| var inputElement = document.getElementById("inputID");//js对象  var $input = $(inputElement);//jquery对象  var txt = $input.val();  alert(txt); |

（4）jQuery对象转成js对象

语法1：jQuery对象[下标，从0开始]

语法2：jQuery对象.get(下标，从0开始)

例如：$div[0]---->divElement

注意：不同的对象只能调用对应的api方法，即jQuery对象不能调用js对象的api，反之亦然

$div.innerHTML（错）

divElement.html(错)

|  |
| --- |
| var $div = $("#divID");//jquery对象  var divElement = $div[0];//js对象(方式一)  //var divElement = $div.get(0);//js对象(方式二)  var txt = divElement.innerHTML;  alert(txt); |

**十）js对象和jQuery对象的区别**

（1）js对象的三种基本定位方式

（A）通过ID属性：document.getElementById()

（B）通过NAME属性：document.getElementsByName()

（C）通过标签名：document.getElementsByTagName()

（2）jQuery对象的三种基本定位方式

（A）通过ID属性：$("#id属性值")

（B）通过标签名：$("标签名")

（C）通过CLASS属性：$(".样式名")

（3）js对象出错的显示

没有合理的提示信息

例如：alert(document.getElementById("usernameIDD").value)

（4）jQuery对象出错的显示

有合理的提示信息，例如：undefined

例如：alert($("#usernameIDD").val())

**十一）jQuery九类选择器（上）【参见jQueryAPI.chm手册】**

目的：通过选择器，能定位web页面（HTML/JSP/XML）中的任何标签

1. 基本选择器【参见selector\_1.html】

|  |
| --- |
| //1)查找ID为"div1ID"的元素个数  alert( $("#div1ID").size() );    //2)查找DIV元素的个数  alert( $("div").size() );    //3)查找所有样式是"myClass"的元素的个数  alert( $(".myClass").size() );    //4)查找所有DIV,SPAN,P元素的个数  alert( $("div,span,p").size() );    //5)查找所有ID为div1ID,CLASS为myClass,P元素的个数  alert( $('#div1ID,.myClass,p').size() ); |

1. 层次选择器【参见selector\_2.html】

|  |
| --- |
| //1)找到表单form里所有的input元素的个数  alert( $("form input").size() );    //2)找到表单form里所有的子级input元素个数  alert( $("form > input").size() );    //3)找到表单form同级第一个input元素的value属性值  alert( $("form + input").val() );    //4)找到所有与表单form同级的input元素个数  alert( $("form ~ input").size() ); |

1. 增强基本选择器【参见selector\_3.html】

|  |
| --- |
| //1)查找UL中第一个元素的内容  alert( $("ul li:first").text() );  //2)查找UL中最后个元素的内容  alert( $("ul li:last").text() );  //4)查找表格的索引号为1、3、5...奇数行个数，索引号从0开始  alert( $("table tr:odd").size() );  //5)查找表格的索引号为2、4、6...偶数行个数，索引号从0开始  alert( $("table tr:even").size() );  //6)查找表格中第二行的内容，从索引号0开始，这是一种祖先 后代 的变化形式  alert( $("table tr td:eq(1)").text() );  //7)查找表格中第二第三行的个数，即索引值是1和2，也就是比0大  alert( $("table tr:gt(0)").size() );  //8)查找表格中第一第二行的个数，即索引值是0和1，也就是比2小  alert( $("table tr:lt(2)").size() );  //9)给页面内所有标题<h1><h2><h3>加上红色背景色，且文字加蓝色  $(":header").css("background-color","red").css("color","blue");    //3)查找所有未选中的input为checkbox的元素个数  alert( $(":checkbox:NOT(:checked)").size() ); |

1. 内容选择器【参见selector\_4.html】

|  |
| --- |
| //1)查找所有包含文本"John"的div元素的个数  alert( $("div:contains('John')").size() );    //2)查找所有p元素为空的元素个数  alert( $("p:empty").size() );    //3)给所有包含p元素的div元素添加一个myClass样式  $("div:has(p)").addClass("myClass");    //4)查找所有含有子元素或者文本的p元素个数,即p为父元素  alert( $("p:parent").size() ); |

。。。

**十二）jQuery中常用方法（上）**

目的：通过方法，能操作web页面（HTML/JSP）中的任何标签

（1）val()：获取标签的value属性值，前提是该标签有value属性

（2）html()：获取标签之间的内容，不能用运于xml文件

（3）text()：获取标签之间的内容，可以用运于html/jsp和xml文件，(提倡)

（4）css()：加key-value形成的css样式

（5）addClass()：加已经定义好的一个css样式

（6）size()：获取jQuery对象/数组中元素的个数，提倡

（7）length：获取jQuery对象/数组中元素的个数

注意：在写jQuery代码时，不光可以使用jQuery的API，还能使用传统JS的API

作业：

1）基于struts2的json插件包，完成省份->城市->区域三级下拉联动

2）完成exe目录中的内容[暂不做]